



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/802,627	03/08/2001	Dean R.E. Long	SUN1P280/P4889	7649
22434	7590	01/25/2005	EXAMINER	
BEYER WEAVER & THOMAS LLP			BANANKHAH, MAJID A	
P.O. BOX 70250			ART UNIT	PAPER NUMBER
OAKLAND, CA 94612-0250			2127	

DATE MAILED: 01/25/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

## Office Action Summary

Application No.

09/802,627

Applicant(s)

LONG ET AL.

Examiner

Majid A Banankhah

Art Unit

2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE \_\_\_\_\_ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 08 March 2001.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 22-40 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 22-40 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 08 March 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- ☒ Notice of References Cited (PTO-892)
- ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.
- ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- ☐ Notice of Informal Patent Application (PTO-152)
- ☐ Other: \_\_\_\_\_.

**DETAILED ACTION**

1. This office action is in response to application, and preliminary Amendment filed on March 08, 2001. Claims 22-40 are presented for examination.

***Claim Rejections - 35 USC § 102***

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

3. Claim 22-40 are rejected under 35 U.S.C. 102(a) as being anticipated by Ole Agesen et al. (An Efficient Meta-Lock for Implementing Ubiquitous Synchronization, Oct. 1999, Sun Microsystems Laboratories).

As a way of claim construction and reference interpretation, the Examiner is preparing the definition of terms from specification, along with the definitions used in the art.

Definition of Terms used in the specification	Definition used in the art.
<b>Lock:</b> Synchronization constructs such as locks [2: 8], Within an object-based system 102, threads 104 may attempt to acquire, or lock, an object 108 [2: 20-21]	<b>Lock:</b> The term refers to the access granted to a particular thread that has entered a synchronized method or a synchronized block. We say that a thread has entered such a method or block has acquired the lock [page 199: <i>lock</i> ], JAVA Threads, by Scott Oaks & Henry Wong, O'REILLY, and Copyright 1999.
<b>Monitor:</b> Synchronization constructs such as	<b>Monitor:</b> A generic synchronization term used

Art Unit: 2127

<p>locks, mutexes, semaphores, and <b>monitors</b> are often used to control access to shared resources [2: 8-9], In some virtual machines, when a thread wants to execute statements which require access to an object, the thread acquires a monitor associated with the object. A monitor, as will be understood by those skilled in the art, a mechanism which is arranged to enable one thread at a time to execute a specific region of code. Hence, <b>a monitor may be considered to be a type of lock</b> [2: 13-17].</p>	<p>in inconsistently between threading systems. In some systems, a monitor is simply a lock; in others, a monitor is similar to the wait and notify mechanism [page 199: <i>monitor</i>], JAVA Threads, by Scott Oaks &amp; Henry Wong, O'REILLY, Copyright]</p>
<p><b>Free list:</b> Freelist 112 is such that an object 108 may access freelist 112 to obtain a monitor 116. That is, object 108 uses freelist 112 to locate a suitable monitor 116 to associate with itself.</p>	<p><b>Free list:</b> each thread has a set of lock records for its exclusive use, linked together for in a free list [pg. 215: section 5.1, 2<sup>nd</sup> para.]. Ole Agesen, et al.</p> <p>Considering monitor to be a type of lock (applicant's definition), and "if an object is not being used by a thread, then the object does not need an associated monitor" (application page 2, last line to page 3 first line), because the thread locks the object, and object obtain the lock, this definition meets applicant's definition.</p>
<p><b>Reference Field:</b> Finally, a reference from the <b>thread</b> substantially directly to the <b>first monitor</b> is set by updating contents of a reference field associated with the thread [6: 21-23].</p> <p>Using the contents of the reference field to identify the first monitor, and updating the contents of a monitor field associated with the first monitor to indicate that the first monitor is in use [6: 30-33].</p> <p>As previously described, thread T1 may set an effective Boolean reference count within itself, e.g., in a reference field, that references the monitor [11: 27--29].</p>	<p><b>Reference Field:</b></p> <p>Event though the reference of Ole Agesen, does not use the phrase "reference field", Ole Agesen, et al. inherently teaches of the reference field in col. 211, left column, under the title "<i>lock record</i>".</p> <p>When a thread acquires the <b>monitor-lock</b>, it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belongs to the <b>thread</b> that holds the monitor lock [211: LFT col., last four lines].</p> <p>Where,</p> <p><b>Monitor-lock:</b> Acquiring and releasing the monitor-lock of an object correspond to entering and exiting the object's monitor. [213: 4.3, locking and unlocking object]</p>

Art Unit: 2127

Based on the above definitions from specifications and the reference of Agesen, the rejection is as follows:

Claim 22: A method for acquiring access to an object in a multi-threaded, object-based system, the method comprising:	The reference of Agesen, [207: R. col. last para.], and [208: L. col. 2.1]
associating a monitor with the object;	A global table called the monitor cache maps object handle addresses to monitor structures [209: L. col. 2 <sup>nd</sup> para.], and <i>lock bits</i> , hold the <i>lock state</i> , meta lock information for the object [211: L. col. 1 <sup>st</sup> para.]
setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object and	Object has header, [210, R. col. <i>thread</i> ], and Fig.2, meta lock information for the object, and header word and the high 30 bits of the multi-use word, and pointer to synchronization data, which thread hold the monitor lock [211, L. col. 1 <sup>st</sup> para]
setting a reference substantially directly from the thread to the monitor so as to enable a garbage collector to determine whether the monitor is suitable for reclamation during a garbage collection process implemented by the garbage collector	<p>Ole Agesen, et al. inherently teaches of the reference in col. 211, left column, 2<sup>nd</sup> para.</p> <p>A lock record represent a thread (on a particular object), monitor lock information, a field for displaced hash and age information (this is garbage collector's age information)[211: L. col.], and when a thread acquires the <b>monitor-lock</b>, it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belong to the <b>thread</b> that holds the monitor lock [211: L. col., last four lines].</p> <p>Which indicates to the garbage collector when the monitor is busy.</p>

Art Unit: 2127

<p>23. A method for acquiring access to an object as recited in claim 22 wherein setting the pointer and setting the reference occur substantially when the thread acquires a lock to the object.</p>	<p>The pointer to synchronization data, indicate which thread owns the monitor lock, which inherently teaches that it happens at the time of thread acquiring the lock [211, L. col. 1<sup>st</sup> para].</p> <p>The reference is also set when the thread acquire the lock [211: L. col., last four lines].</p>
<p>24. A method for acquiring access to an object as recited in claim 22 wherein setting the reference includes setting a Boolean reference count.</p>	<p>The two least significant bits are used to show lock states therefore they are Boolean expression [211: L. col. 1<sup>st</sup> para].</p>
<p>25. A method for acquiring access to an object as recited in claim 22 wherein associating the monitor with the object includes setting a first pointer from the monitor to the object.</p>	<p>and <i>lock bits</i>; hold the <i>lock state</i>, meta lock information for the object [211: L. col. 1<sup>st</sup> para.], also Fig.2.</p>
<p>26. A method for acquiring access to an object as recited in claim 22 wherein associating a monitor with the object includes obtaining the monitor from a free-list of monitors.</p>	<p>each thread has a set of lock records for its exclusive use, linked together for in a free list [pg. 215: section 5.1, 2<sup>nd</sup> para.]. Ole Agesen, et al.</p>
<p>27. A method for acquiring access to an object as recited in claim 22 wherein associating the monitor with the object includes setting a second pointer from the object to the monitor.</p>	<p>The reminder of the multi-use lock points to a lock record [pg. 211: L. col., 1st para.]. Ole Agesen, et al.</p>

<p>28. A method for reducing overhead associated with providing a monitor for an object included in a multi-threaded, object-based computing system, the computing system including a plurality of monitors which includes the first monitor, the method comprising:</p>	<p>Agesen, [207, L. R. col. 1<sup>st</sup> para. multi-threaded], [208, L. col. 2.1, object based], and [208: L. col. 2.1, 2<sup>nd</sup> para. wait notify, and notify all operations], and 209: L. col. 2<sup>nd</sup> para. global table called the monitor cache and monitor structures].</p>
--	---

Art Unit: 2127

setting a first pointer from the first monitor to the object;	and <i>lock bits</i> ; hold the <i>lock state</i> , meta lock information for the object [211: L. col. 1 <sup>st</sup> para.], also Fig.2.
setting a second pointer from the object to the first monitor;	The reminder of the multi-use lock points to a lock record [pg. 211: L. col., 1st para.]. Ole Agesen, et al.
setting a third pointer from a thread to the object, wherein the thread is arranged to execute on the object; and	Object has header, [210, R. col. <i>thread</i> ], and Fig.2, meta lock information for the object, and header word and the high 30 bits of the multi-use word, and pointer to synchronization data, which thread hold the monitor lock [211, L. col. 1 <sup>st</sup> para]
setting a reference substantially directly from the thread to the first monitor, wherein setting the reference includes updating contents of a reference field associated with the thread.	Ole Agesen, et al. inherently teaches of the reference in col. 211, left column, 2 <sup>nd</sup> para.  A lock record represent a thread (on a particular object), monitor lock information, a field for displaced hash and age information (this is garbage collector's age information)[211: L. col.], and when a thread acquires the <b>monitor-lock</b> , it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belong to the <b>thread</b> that holds the monitor lock [211: L. col., last four lines].  Which indicates to the garbage collector when the monitor is busy.

29. A method for reducing overhead as recited in claim 28 further including invoking the object using the thread, wherein invoking the object using the thread includes initiating a wait action, the wait action being arranged to place the object into a wait state.	The <i>WATERS</i> state, the object is no longer monitor locked [211: L. col.], thread attempting to lock object reads the object's multi-use word [215: R. col. 5.2, first para.]
---	--

Art Unit: 2127

30. A method for reducing overhead as recited in claim 29 further including, performing memory reclamation during the wait state.	Periodically unused monitors are reclaimed [215: L. col. 5.1, first para.]
31. A method for reducing overhead as recited in claim 30 wherein performing a memory reclamation during the wait state includes: identifying the contents of the reference field;	Periodically unused monitors are reclaimed [215: L. col. 5.1, first para.], see also Definition of reference field [ <i>supra</i> ], without identifying the content of the reference field the next operation (below) would be impossible.
using the contents of the reference field to identify the first monitor, and updating the contents of a monitor field associated with the first monitor to indicate that the first monitor is in use.	When a thread acquires the <b>monitor-lock</b> , it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belongs to the <b>thread</b> that holds the monitor lock [211: LFT col., last four lines]. Where, <b>Monitor-lock:</b> Acquiring and releasing the monitor-lock of an object correspond to entering and exiting the object's monitor. [213: 4.3, locking and unlocking object]
32. A method for reducing overhead as recited in claim 31 wherein performing a memory reclamation during the wait state further includes:  scanning through substantially all monitors included in the plurality of monitors; and reclaiming substantially any monitor included in the plurality of monitors that is not indicated as being in use, wherein the first monitor is not reclaimed because the contents of the monitor field associated with the first monitor indicate that the first monitor is in use.	Periodically unused monitors are reclaimed [215: L. col. 5.1, first para.]  When a thread acquires the <b>monitor-lock</b> , it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belongs to the <b>thread</b> that holds the monitor lock [211: LFT col., last four lines]. Where, <b>Monitor-lock:</b> Acquiring and releasing the monitor-lock of an object correspond to entering and exiting the object's monitor. [213: 4.3, locking and unlocking object]  Sine reclaiming step, uses unused monitors, the scanning step is inherent.
33. A computer program product for acquiring access to an object in a multi-threaded, object-based system, the computer program product comprising: computer code for associating a monitor with the object;	A global table called the monitor cache maps object handle addresses to monitor structures [209: L. col. 2 <sup>nd</sup> para.], and <i>lock bits</i> , hold the <i>lock state</i> , meta lock information for the object [211: L. col. 1 <sup>st</sup> para.]



Art Unit: 2127

computer code for setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object;	Object has header, [210, R. col. <i>thread</i> ], and Fig.2, meta lock information for the object, and header word and the high 30 bits of the multi-use word, and pointer to synchronization data, which thread hold the monitor lock [211, L. col. 1 <sup>st</sup> para]
computer code for setting a reference substantially directly from the thread to the monitor so as to enable a garbage collector to determine whether the monitor is suitable for reclamation during a garbage collection process implemented by the garbage collector; and a computer-readable medium that stores the computer codes.	<p>Ole Agesen, et al. inherently teaches of the reference in col. 211, left column, 2<sup>nd</sup> para.</p> <p>A lock record represent a thread (on a particular object), monitor lock information, a field for displaced hash and age information (this is garbage collector's age information)[211: L. col.], and when a thread acquires the <b>monitor-lock</b>, it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belong to the <b>thread</b> that holds the monitor lock [211: L. col., last four lines].</p> <p>Which indicates to the garbage collector when the monitor is busy.</p>
34. A computer program product according to claim 33 wherein the computer code for setting the reference includes computer code for setting a Boolean reference count.	The two least significant bits are used to show lock states therefore they are Boolean expression [211: L. col. 1 <sup>st</sup> para].
35. A computer program product according to claim 33 wherein the computer-readable medium is one selected from the group consisting of a hard disk, a floppy disk, a data signal embodied in a carrier wave, a tape drive, an optical drive, and a CD-ROM.	Computer program product of Agesen inherently must be stored on some media such as hard disk, a floppy disk, a data signal embodied in a carrier wave, a tape drive, an optical drive, and a CD-ROM.

36. A computer program product for reducing overhead associated with providing a monitor for an object included in a multi-threaded,	Agesen, [207, L. R. col. 1 <sup>st</sup> para. multi-threaded], [208, L. col. 2.1, object based], and [208: L. col. 2.1, 2 <sup>nd</sup> para. wait notify, and
--	---

Art Unit: 2127

object-based computing system, the computing system including a plurality of monitors which includes the first monitor, the computer program product comprising:	notify all operations], and 209: L. col. 2 <sup>nd</sup> para, global table called the monitor cache and monitor structures].
computer code for setting a first pointer from the first monitor to the object,	and <i>lock bits</i> ; hold the <i>lock state</i> , meta lock information for the object [211: L. col. 1 <sup>st</sup> para.], also Fig.2.
computer code for setting a second pointer from the object to the first monitor;	The reminder of the multi-use lock points to a lock record [pg. 211: L. col., 1st para.]. Ole Agesen, et al.
computer code for setting a pointer from a thread to the object, wherein the thread is arranged to execute on the object;	Object has header, [210, R. col. <i>thread</i> ], and Fig.2, meta lock information for the object, and header word and the high 30 bits of the multi-use word, and pointer to synchronization data, which thread hold the monitor lock [211, L. col. 1 <sup>st</sup> para]
computer code for setting a reference substantially directly from the thread to the first monitor, wherein setting the reference include updating contents of a reference field associated with the thread; and a computer-readable medium that stores the computer codes.	Ole Agesen, et al. inherently teaches of the reference in col. 211, left column, 2 <sup>nd</sup> para.  A lock record represent a thread (on a particular object), monitor lock information, a field for displaced hash and age information (this is garbage collector's age information)[211: L. col.], and when a thread acquires the <b>monitor-lock</b> , it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belong to the <b>thread</b> that holds the monitor lock [211: L. col., last four lines].  Which indicates to the garbage collector when the monitor is busy.
37. A computer program product according to claim 36 further including: computer code for invoking the object using the thread, wherein invoking the object using the thread includes initiating a wait action, the wait action being arranged to place the object into a wait state, and	The <i>WATERS</i> state, the object is no longer monitor locked [211: L. col.], thread attempting to lock object reads the object's multi-use word [215: R. col. 5.2, first para.]

Art Unit: 2127

computer code for performing memory reclamation during the wait state.	Periodically unused monitors are reclaimed [215: L. col. 5.1, first para.].
38. A computer program product according to claim 37 wherein the computer code for performing a memory reclamation during the wait state includes: computer code for identifying the contents of the reference field;	Periodically unused monitors are reclaimed [215: L. col. 5.1, first para.], see also Definition of reference field [ <i>supra</i> ], when a thread acquires the <b>monitor-lock</b> , it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belongs to the <b>thread</b> that holds the monitor lock [211: L. col., last four lines].
computer code for using the contents of the reference field to identify the first monitor;	Without identifying the first monitor the previous operations would be impossible.
computer code for updating the contents of a monitor field associated with the first monitor to indicate that the first monitor is in use	New lock record are appended to the end of the queue and stay in order, except that when a thread acquires the monitor lock, it moves its lock record to the front so that the first lock record of a first lock object always belong to the thread that holds the monitor lock [211, L. col., last para.]
computer code for scanning through substantially all monitors included in the plurality of monitors; and	Sine reclaiming step, uses unused monitors, the scanning step is inherent [Periodically unused monitors are reclaimed [215: L. col. 5.1, first para.]]
computer code for reclaiming substantially any monitor included in the plurality of monitors that is not indicated as being in use, wherein the first monitor is not reclaimed because the contents of the monitor field associated with the first monitor indicate that the first monitor is in use.	When a thread acquires the <b>monitor-lock</b> , it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belongs to the <b>thread</b> that holds the monitor lock [211: LFT col., last four lines]. Where,
39. A computer program product according to claim 36 wherein the computer-readable medium is one selected from the group consisting of a hard disk, a Floppy disk, a data signal embodied in a carrier wave, a tape drive, an optical drive,	Computer program product of Agesen inherently must be stored on some media such as hard disk, a floppy disk, a data signal embodied in a carrier wave, a tape drive, an optical drive, and a CD-ROM.

Art Unit: 2127

and a CD-ROM.	
40. A multi-threaded, object-based computing system comprising: an object; a thread that includes a first reference to the object; and	Object has header, [210, R. col. <i>thread</i> ], and Fig.2, meta lock information for the object, and header word and the high 30 bits of the multi-use word, and pointer to synchronization data, which thread hold the monitor lock [211, L. col. 1 <sup>st</sup> para]
a monitor that includes a second reference to the object,  wherein the object includes a third reference to the monitor  and the thread includes a fourth reference to the monitor  so as to enable a garbage collector to determine whether the monitor is suitable for reclamation during a garbage collection process implemented by the garbage collector.	and <i>lock bits</i> ; hold the <i>lock state</i> , meta lock information for the object [211: L. col. 1 <sup>st</sup> para.], also Fig.2.  The reminder of the multi-use lock points to a lock record [pg. 211: L. col., 1st para.]. Ole Agesen, et al.  Ole Agesen, et al. inherently teaches of the reference in col. 211, left column, 2 <sup>nd</sup> para.  A lock record represent a thread (on a particular object), monitor lock information, a field for displaced hash and age information (this is garbage collector's age information)[211: L. col.], and when a thread acquires the <b>monitor-lock</b> , it moves its lock record to the front so that the <b>first lock record</b> of the locked object always belong to the <b>thread</b> that holds the monitor lock [211: L. col., last four lines].  Which indicates to the garbage collector when the monitor is busy, and Periodically unused monitors are reclaimed [215: L. col. 5.1, first para.].

Art Unit: 2127

### Conclusion

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Majid A. Banankhah** whose telephone number is (571) 272-3770. The examiner can normally be reached on Monday – Thursday, 8:00 AM – 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756.

Information regarding the status of an application may be obtained from the patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll free).

Maid A. Banankhah

1/13/05

MAJID BANANKHAH  
PRIMARY EXAMINER  
